React beginner

Let’s mix some HTML and JS together

Red Hat

Agenda

Project update - design system and usage

What is React? Definition

Props manipulating

Component lifecycle

Render cycle - when rendering happens

Different data binding (one vs two way)

Class vs function component

Landing and dashboard

Project update - design system and usage

What is React? Definition

- Declarative, component based Javascript library for building UI
  - Components are (sometimes) stand alone units with well defined API
  - You don’t need to know how the component works under the hood to be able to use it
- IS NOT state management library
- IS NOT data fetching library
- IS NOT data visualization library
- IS NOT framework
- React is build on rather simple principles which allow developers build complex solutions to satisfy their customers needs
- You can implement simple version of React in a matter of minutes
  - https://www.youtube.com/watch?v=f2mMOiCSj5c

https://reactjs.org/

Component and props

- Component is a stand alone UI piece with well defined API
- API can provide a set of props
  - Props is an object
  - You can send multiple props to single component
  - Key is a prop name, and value is its value
- Prop is a variable which is consumed and used by the component
- Props are given to component from its parent
- Prop can be required or optional
- Component will most likely throw an error if its missing a required props
- Prop requirement and props can be defined via propTypes
- PropTypes is a static attribute on a component type
- Each prop type definition is a function that checks the prop content

Props manipulating

- Two types of props
  - Values - passing data down
  - Callbacks - user (or async) interactions
- Use spread to control multiple props
  - Combine them together
  - Spread them to pass them down

```
const InputComponent = ({ onChange, ...props }) => {
 return (
   <input
    { ...props }
     onChange={(event) =>
onChange(event.target.value)}
   />
 )
}
```

Component props examples

```
const TitleComponent = props => {
 const {text, size} = props;
 return (
   <h1 style={{ fontSize: size }}>{text}</h1>
 )
};

const Parent = () => {
 return (
   <TitleComponent text="Hello" size={48} />
 )
}
```

Component prop types examples

```
import PropTypes from 'prop-types';
const TitleComponent = props => {
 const {text, size} = props;
 return (
   <h1 style={{ fontSize: size }}>{text}</h1>
 )
};

TitleComponent.propTypes = {
 text: PropTypes.string.isRequired, // will throw an error if the prop missing
 size: PropTypes.number // will be set to 24 if the prop is missing
};

TitleComponent.defaultProps = {
 size: 24
};
```

Component and state

- Component can have its internal state
  - Should not hold application state (data)
    - You have state management libraries for that
  - State should only hold component specific data
  - What is the value of input? Is the dropdown expanded?
  - ~~What are the attributes of value logged in user?~~
- State is directly accessible only within the component
- If you want to expose it to children, it must be sent to them as a prop

Component state example

```
const InputComponent = () => {
 const [value, setValue] = useState('');
 const handleInputChange = (({ target: { value } }) => setValue(value));
 return (
   <input
       name="controlled-input"
       value={value}
       placeholder="No value"
       onChange={handleInputChange}
    />
 )
};
```

Component lifecycle

- After React dev team introduced hooks, the life cycles of react component has changed significantly
  - Preparation for concurrent mode and async rendering
- You have different life cycles for React **Class** components and **Functional** components

https://reactjs.org/docs/state-and-lifecycle.html#adding-lifecycle-methods-to-a-class

Component lifecycle – Classes

There are several predefined class functions with a static trigger sequence

- constructor
  - Is triggered when the class is instantiated
  - Useful for setting initial component state, registering listeners...
  - In most cases is not really necessary
- componentWillMount (deprecated)
  - After constructor, but before first render
  - Was used for initial data fetch
  - Misuse will cause faulty state changes in async mode
- componentDidMount
  - Is called **once** only after **initial render**
  - New place for initial data fetch

https://reactjs.org/docs/state-and-lifecycle.html#adding-lifecycle-methods-to-a-class

Component lifecycle - Classes

There are several predefined class functions with a static trigger sequence

- componentWillReceiveProps (deprecated)
  - Component is about to receive new props, but it did not re-render
- componentDidUpdate
  - Props or component internal state were updated
  - Based you can compare new and previous props/state to make some updates, api calls, logs, etc.
  - Danger of infinite loops
  - State changes must be **wrapped in condition**
- componentWillUnmount
  - Clean up phase before the component is removed from virtualDOM

https://reactjs.org/docs/state-and-lifecycle.html#adding-lifecycle-methods-to-a-class

Component lifecycle – Classes

There are several predefined class functions with a static trigger sequence

- shouldComponentUpdate
  - Rending is the most expensive operation in DOM an any library/framework
  - Developer can use this method to programmatically check whether to trigger render method or not
  - This will affect component children and not send new props to them
  - If children rely on parent component you cannot use this in most cases (lifting props to app state)
  - May have major performance benefits on your app

https://reactjs.org/docs/state-and-lifecycle.html#adding-lifecycle-methods-to-a-class

Component lifecycle – Classes

There are several predefined class functions with a static trigger sequence

- Render
  - Main rendering method
  - **Must be implemented** in every class component and return something that can be rendered
  - Render is triggered on prop and state changes by default
    - Can be handled via shouldComponentUpdate
- componentDidCatch
  - Handler for unpredicted errors in virtual DOM
  - The “Whoops! Something has happened.” screen

https://reactjs.org/docs/state-and-lifecycle.html#adding-lifecycle-methods-to-a-class

Component lifecycle – functions

You can handle functional component lifecycle via **useEffect** hook

- Possible in React version 16.8.x and later
- Allows performing **side effects** in your functions
  - Side effect triggers something outside of a function scope
  - Breaks the pure function rule - same input may give you different output
  - Necessary for efficiently reacting on (user) events
- Can replace all lifecycle methods, except componentDidCatch

https://reactjs.org/docs/hooks-intro.html

Component lifecycle - functions

Use effect has this API

```
const Component = ({ username }) => {
 useEffect(() => {
   API.getUserDataAndUpdateAppGlobalState(`api/user?username=${username}`)
   return () => {
     cleanAppGlobalState()
   }
 }, [username])
 return (
   <h1>{username}</h1>
 )
}
```


https://reactjs.org/docs/hooks-intro.html

Component lifecycle – functions

Arguments

- Effect function
- List of triggers (dependencies)

There can be multiple effects, reacting on different triggers.

That way, we can mimic the life cycles of classes.

```
const Component = ({ username }) => {
 useEffect(() => {
   API.getUserDataAndUpdateAppGlobalState(`api/…`
   return () => {
     cleanAppGlobalState()
   }
 }, [username])
 return (
   <h1>{username}</h1>
 )
}
```


https://reactjs.org/docs/hooks-intro.html

Component lifecycle – functions

- No list of triggers means that the effect will trigger on every props/state change
- Will cause infinite loop if state is changed here
- Same use cases as **componentDidUpdate**
- Will also trigger like **componentDidMount**

```
useEffect(() => {
  /**effect */
})
```


https://reactjs.org/docs/hooks-intro.html

Component lifecycle – functions

- Empty list mean that it will trigger only once, after initial mount
- **componentDidMount**
- Does not react on any props/state updates

```
useEffect(() => {
  /**effect */
}, [])
```


https://reactjs.org/docs/hooks-intro.html

Component lifecycle – functions

- Effect will be called when any variable in the list is changes
- **Non primitive types must follow immutable pattern to trigger effect -> Must return new instance**
- Also will be triggered in first render
- **componentDidUpdate**, **componentDidMount**

```
useEffect(() => {
  /**effect */
}, [propName, stateName])
```


https://reactjs.org/docs/hooks-intro.html

Component lifecycle – functions

- If effect returns a function, it will be called before component is unmounted from DOM
- **componentWillUnmount**

```
useEffect(() => {
  /**effect */
  return () => {
    /**clean up */
  }

}, [propName, stateName])
```


https://reactjs.org/docs/hooks-intro.html

Render cycle – when rendering happens

- Render function is triggered when
  - Component props has changed
  - Component state has changed
  - Component context has changed
  - Parent has re-rendered
- Everything can be optimized/block to save rendering cycles
- Render will not trigger if props/state are **MUTATED VARIABLES OF THE SAME INSTANCE**
  - Examples will follow

Different data binding (one vs two way)

- One way data binding
  - Data can be send only in one direction in the DOM
  - From parents to children (React)
  - From children to parents (not a good idea)
- Two way data binding
  - Data can be accessed anywhere
  - You (component) have access to your children and parent data

One way data binding

- Used by all modern UI libraries/frameworks
- Data is “bubbling” down through nodes to the leaves of the DOM tree
- Predictable behaviour
- Forces component independence
- If component wants to consume some data, it must provide an API through which it can receive such data
- Although technically parents can access its children data it's not a good idea
    - **DO NOT TOUCH CHILDREN**!
    - It's not a joke, it's not a really good idea to access children data even though you can
    - It opens pandora's box of bugs

Two way data binding

- Used in older libraries/frameworks
- One of the reasons why original Angular was abandoned
- Developers ignored good practices and were accessing parent data from children
- Components lost their independence
- UI can become rigid and entangled into horrific spaghetti
- In order to “fix” such entanglement, the code became on big switch statement

```
const InputComponent = () => {
 return (
   <input
     name="controlled-input"
     value={parent.parent.data.value}
     placeholder="No value"
     onChange={parent.parent.parent.parent.handleInputChange}
   />
 )
}
```

Class vs function component – Function components

- Components are state and props in a nutshell
- React is moving towards function component
  - Hooks to control lifecycle
  - Hooks to connect to redux and router
  - Hooks to control everything
- Prefer functions for calculation to be moved out of function component as they are reinitialized on every render

```
const InputComponent = () => {
 return (
   <input />
 )
}
```

Class vs function component - Class component

- Some hooks are still missing so classes will still be around
- Binding of this in classes is necessary
- To ease binding @babel/plugin-proposal-class-properties should be used
- Do not call setState from render method as you end up with infinit loop

```
class InputComponent extends Component {
 render() {
  return (
   <input />
  )
 }
}
```

Optimization hooks

- There are several hooks that can help you optimize your code
  - As always, before optimizing, check if you actually need it
- useCallback
  - If you need some function to be initialized within render cycle (usually for debounce)
- useMemo
  - If you need some value to be calculated on each render and you don’t want to change it that often (for instance when you want to build pagination)
- useRef
  - Special hook, that behaves similar to useState, but does not trigger re-render

Additional hooks

- useContext
  - React.provider hook to consume context
  - Easy to use multiple context providers in one component
- useReducer
  - State management hook, if you need to store big chunk of data in component
- Custom hooks
  - You can write your own hooks, and share them in your library
  - Great examples: useDispatch, useSelector from react-redux library
  - Any function can be “hook” as long as it uses any React’s hooks

Redux advanced

Thinking with redux - one state to rule them all

Red Hat

Agenda

What is redux - principles

Going from action type trough action to state update

How to use connect - 3 parts of connect function

How to use hooks with redux

Middlewares and other cool tricks (immutable, reselect)

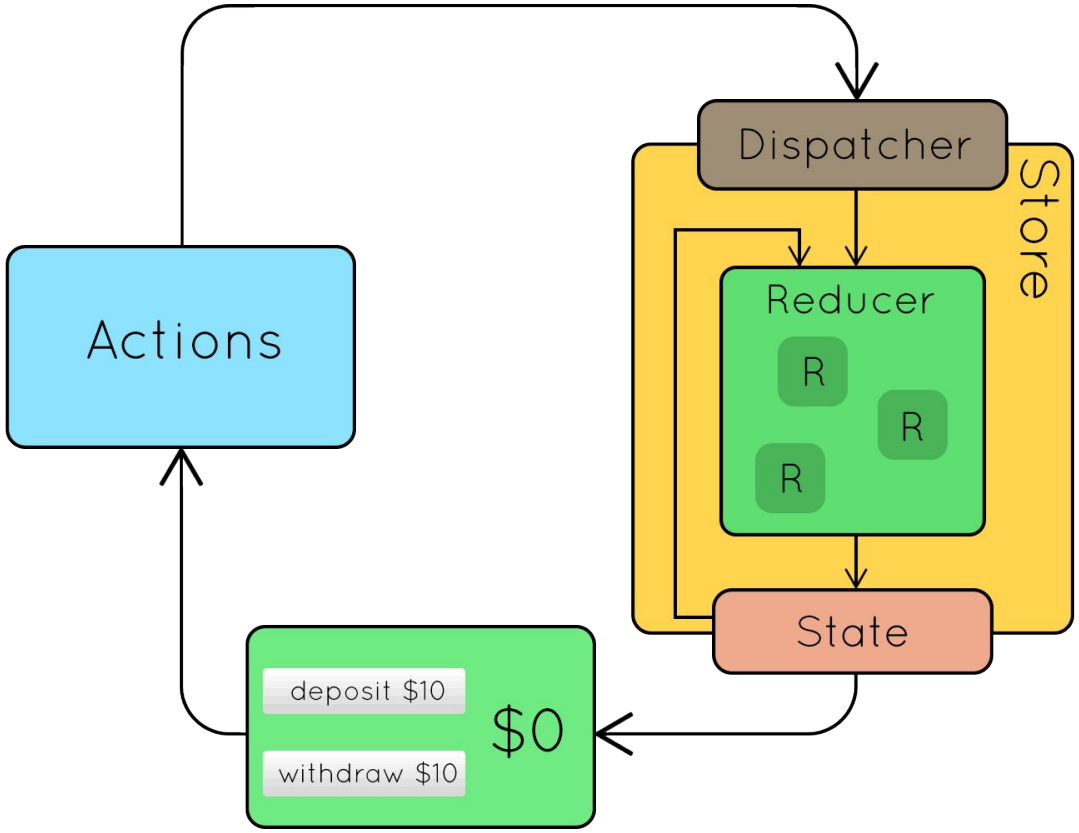
You might not need redux - usereducer

Let’s do some coding

What is redux – definition

- A Predictable State Container for JS Apps
- Something triggers action - pure function that returns object
- Action is dispatched into store
- Reducers pass this action around and mutate state based on type and payload
- You can track history of actions because they are pure functions

What is redux – principles



https://github.com/hmajid2301/articles/tree/master/10.%20Redux%20with%20React%20Navigation

Going from action type

- Action type - string constant to identify action, kinda like name

```
export const SOME_ACTION = 'SOME_ACTION_TYPE';
```

Going from action type trough action

- Action (flux style action) - combined type, payload, error, meta
  - Type - to identify action
  - Payload (optional) - actual data, usually object
  - Meta (optional) - additional data, usually to identify records
  - Error (optional) - boolean value to indicate error

```
const doSomeAction = (data, entityId) => ({
   type: SOME_ACTION,
   payload: {
       entityId,
       data
   },
   meta: { entityId },
   error: false
});
```

https://github.com/redux-utilities/flux-standard-action

Going from action type trough action to state update

- Reducer is function that takes state and action
- Reacts to action and mutates state in expected way

```
const reducer = (state, action) => {
    return {
        ...state,
        entities: state.entities.map(item => ({
            ...item,
            ...item.id === action.entityId && action.data
        }))
    };
};
```

Going from action type trough action to state update and use them

- createStore - to use reducers in a store
  - Reducer function
  - Default state
  - Enhancers
- combineReducers - to namespace your state and split reducers
- applyMiddleware - enhancer function to user middlewares (logger, async functions,etc.)

```
createStore(
 combineReducers({ appState }),
 { appState: {} },
 applyMiddleware(logger)
);
```

How to use redux in react app

- Connect function - old way
  - mapStateToProps
  - mapDispatchToProps (optional)
  - mergeProps (optional)
- Hooks
  - useDispatch
  - useSelector
  - useStore

How to use connect - mapStateToProps

- First argument of connect, it's required
- Store as first argument and component's props as second argument
- Used to take certain part of state in order to limit number of renders

```
const mapStateToProps = ({ entities }) => ({
   rows: entities && entities.rows,
   loaded: entities && entities.loaded,
   selected: entities && entities.selected
});
```

How to use connect - mapDispatchToProps

- Second argument of connect function, it's optional
- Function with dispatch and component's props as attributes
- Returns object with wrapped action functions

```
const mapDispatchToProps = (dispatch) => {
   return {
        loadEntity: (id) => dispatch(actions.loadEntity(id)),
        clearNotifications: () => dispatch(actions.clearNotifications()),
        addNotification: (payload) => dispatch(addNotification(payload)),
        onSelectRows: (id, isSelected) => dispatch(actions.selectEntity(id, isSelected))
  };
}
```

How to use connect – mergeProps

- Function to combine state, dispatch and component's props
- By default it will be used as `{ ...ownProps, ...stateProps, ...dispatchProps }`

```
const mergeProps = (propsFromState, propsFromDispatch, ownProps) => {
 return {
   ...ownProps,
   ...propsFromState,
   ...propsFromDispatch,
   fetchRoles: (apiProps) => propsFromDispatch.fetchRoles(propsFromState.groupId,
apiProps)
 };
};
```

How to use hooks with redux – useSelector

- Replaces mapStateToProps and mergeProps functions
- Allows to pluck pieces of state from store
- It’s recommended to use multiple selectors in component to improve performance
  - Always try to requests only primitive values, not whole objects (not always possible)

```
const Component = ({ userId }) => {
  const name = useSelector(({ userReducer: { profile: { name } } }) => name);
  const email = useSelector(({ userReducer: { profile: { email } } }) => email);
  const orders = useSelector(({ orderReducer: { orders } }) => orders.find(order => order.userId === userId));

  return (
    <div>...</div>
  )
}
```

How to use hooks with redux – useDispatch

- Replaces mapDispatchToProps
- Returns a dispatch function from closest store
- Use this function to call redux actions

```
const Component = () => {
  const dispatch = useDispatch()
  const handleThemeToggle = themeVariant => dispatch({ type: 'TOGGLE_THEME', payload: themeVariant })

  return (
    <div>
      <button onClick={() => handleThemeToggle('blue')}>Change UI to blue theme</button>
      <button onClick={() => handleThemeToggle('orange')}>Change UI to orange theme</button>
    </div>
  )
}
```

How to use hooks with redux – useStore

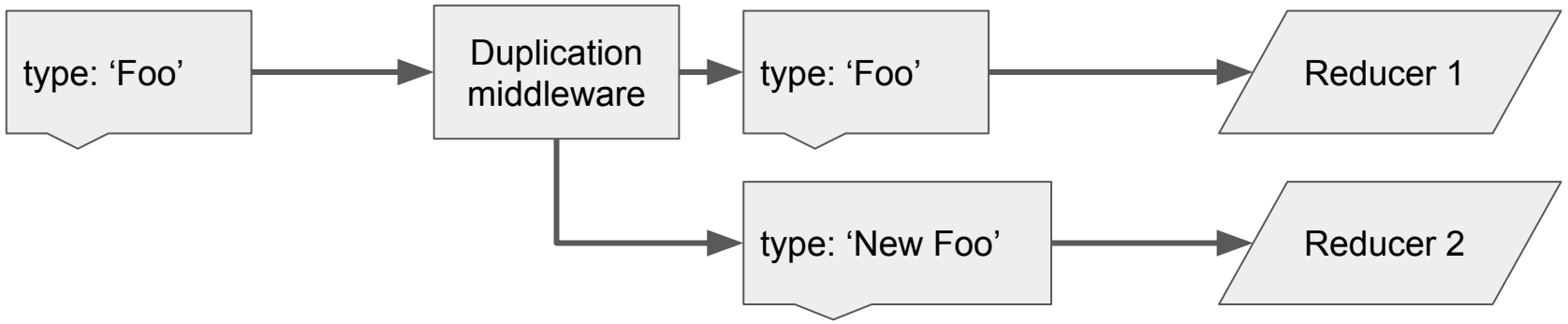
- In order to access store

```
const Component = () => {
 const store = useStore();

 console.log(store);
 return 'FooBar';
}
```

Middlewares and other cool tricks

- As reducer, listens on actions, but catches them before they are passed to reducers
- Can observe, modify action or even prevent it from reaching reducers
- Usually middleware is used to add some side effect to action



You might not need redux – useReducer

- Hook introduced in React version 16.8.0
- Introduces reducers to core React
- Its meant to be used for complex component state updates
  - More than two “setState” calls in one callback
  - Every setState triggers one render always
  - Multiple setState have negative performance impact
- useReducer is here to prevent developers store objects in state (useState)
  - Trigger unnecessary re-renders
- useReducer on its own cannot replace redux
  - Lacks optimizations, middlewares, namespacing, context, etc.
  - Would require additional functions to fully replace redux
  - But at that point, you have implemented redux library
- On its own (with clever context and memo usage), can replace redux in smaller scale applications

Let’s do some coding

- State vs Redux vs UseReducer - https://codesandbox.io/s/friendly-lovelace-sf7s59

Landing and dashboard – let’s write some code

- Old
  - jQuery
  - Ajax
  - Single Session
  - Variables
  - IE drove it in the past
  - Lodash
  - Callback hell
  - Bootstrap
- New
  - Transpilers
  - JS is slowly moving towards functional programming - map, lambda, reduce, etc.
  - SSO with JWT
  - Improved Browser API
  - Dependency hell
  - Build tools - module based codebase
  - Multiple free and opensource design systems
  - Virtual DOM, component architecture, pure functions - React

Single sign on/out/only

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Popis problemu

- Two scenarios
  - Single page web application
  - Library to used in any further project
- All build processes managed from one place
  - Easy use once we have everything all set up
- We want to use the newest javascript goodies
- Some framework so we have easier DOM manipulation
- One package manager

Historie – aneb pred vice nez 5 lety

- Bower was the main place to go for packages
  - It is slowly dying and on 02 Oct 2017 was announced that users should migrate to something else
- First there was Grunt
  - On 11 January 2012 this first task runner was introduced, it was a breakthrough in frontend development.
  - Pros: Fairly simple, easy to use
  - Cons: Temporary files, piping one file trough multiple tasks was cumbersome
- Then there was Gulp.js
  - On 26 September 2013 was released taks runner that was heavily inspired by Grunt
  - Pros: Faster, file piping
  - Cons: Config can be really confusing to newcomers
- jQuery
  - Most hated and loved library to manipulate DOM

https://bower.io/blog/2017/how-to-migrate-away-from-bower/

Proc to selhalo

- NPM took over because it simply had a lot more packages
- NPM also introduced several built in scripts, that ca be ran at certain point of installation/build
- Package managers now contain task runner
- Configs were too hard to maintain - few people really understood file piping
- Debugging of build process was not so easy
- Modern build tools have much better features
    - HTTP/2 code splitting
    - Built in minification
    - Built in sourcemaps
    - Tree shaking
    - And many more

Dnesni stav – package manager / task runner

- Yarn
  - Developed by Facebook
  - Supports workspaces - great for monorepos
  - Easier task runner and package installation
  - First to introduce lock file
- NPM
  - Developed by Google
  - Comes with node
  - Contains NPX
  - Convenient for larger teams spread across globe
- PNPN
  - Honorable mention - hard linking packages instead of downloading and installing

Dnesni stav – build nastroje

- Webpack
  - All purpose build tool
  - Extensive configuration options
  - Configuration is required
  - Extensible
- Parcel
  - Zero config file
- Rollup
  - Focus on building ES libraries

Nutnost transpileru, co to je?

- Babel
    - Javascript transpiler to provide backwards compatibility of new JS syntax quirks and fashion
    - Each file is parsed separately and compiled into native JavaScript
    - Huge collection of plugins - you can even write your own plugins
- Typescript
    - JavaScript flavor/superset with data types
    - Compiles into JavaScript
    - Custom plugins called Language Service Plugin - not that used as with babel

Babel config

- Multiple formats
    - JS, JSON, ES6
- Option to write custom functions in plugins [1]
- `BABEL_ENV` to use different config per different environment
- Option to write config in package.json

```
{
  "presets": [
    [
      "@babel/env",
      {
        "targets": "> 0.25%, not dead"
      }
    ],
    "@babel/react",
  ],
  "plugins": [
    [
      "@babel/plugin-proposal-decorators",
      {
        "legacy": true
      }
    ],
    "babel-plugin-lodash",
    "@babel/plugin-proposal-class-properties"
  ]
}
```

[1] https://www.npmjs.com/package/babel-plugin-import

Scenar: Webova aplikace - Parcel vs Webpack

- React application
- Dev server
    - Proxy to API
- HTML template
- Tree shaking
- One config to rule them all (dev vs prod)
- Custom envs from terminal/.env file
- Static files loader - images, styles

Scenar: Knihovna - Rollup vs Webpack

- React library
- Bundled to one file and to multiple files as well
- Externals
  - 3rd party libraries not bundled in final build
- Named vs Default exports
- Multiple environments
  - ESM, UMD, CommonJs

Conclusion library

Webpack

- No native support for full ESM bundles
  - Webpack 5 promises to implement it
- Harder to set-up
- Easier to read

Rollup

- Everything has to be done through plugins +/-
- Smaller bundle size
- Supports native ESM!
- Named exports
  - You need to list all named exports

Conclusion - funny

- Webpack is like VIM
    - Hard to setup and if something breaks you have to really understand it
    - But the things that can be done with it are almost limitless
- Parcel is like MS word
    - No need to configure anything
    - If you need to tweak something up, it's really hard to do
- Rollup is like VS Code
    - One of the greatest software for one thing
    - Handles poorly web apps

Many people experience our brand by seeing one of the thousands of presentations Red Hatters deliver each year.

From Summit keynotes to conference-room sales meetings, we want our public face to be coherent and recognizable. Our content must be meaningful and relevant to our audiences. Our stories should be told in a clear, compelling way.

How to build an effective presentation
https://pnt.redhat.com/pnt/p-611879/

Getting started with Google Slides
https://gsuite.google.com/learning-center/products/slides/get-started/#!/

Red Hat brand standards
https://www.redhat.com/en/about/brand/standards

Red Hat brand assets
https://pnt.redhat.com/pnt/b-420952/Brand_assets

Red Hat

QUICK TIP
Update or remove the confidential designator on the master slide.

Updating the confidential designator:

Update the designator in Google Slides by choosing “Slide,” then “Edit Master.”
Copy the appropriate designation into the “Confidential designator” field in the upper right.

Red Hat associates only

Change the designator on the master slide to:

CONFIDENTIAL Red Hat associates only

Use this designation for a confidential presentation that can only be shared with Red Hat associates. The Red Hat associate(s) who receive this deck can share it with other Red Hat associate(s), but no one else.

Red Hat associates only, no further distribution

Change the designator on the master slide to:

CONFIDENTIAL Red Hat associates only, No further distribution

Use this designation for a confidential presentation intended only for the Red Hat associate(s) who receive it originally. The Red Hat associate(s) who receive this presentation cannot share it with anyone—inside or outside of Red Hat.

Red Hat associate and NDA partner use only, no further distribution

Change the designator on the master slide to:

CONFIDENTIAL Red Hat associate and NDA partner use only, No further distribution

Use this designation for a confidential presentation intended only for the Red Hat associate(s) and partner(s) with signed NDA who receive the deck originally. The Red Hat and NDA partner associate(s) who receive this deck cannot share it with anyone—even other Red Hat and partner associate(s).

QUICK TIP
Update or remove the project number, event name or hashtag on the master slide.

Red Hat

This section includes:

Title slide templates

Closing slide templates

Divider slide templates

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat

Red Hat Enterprise Linux

QUICK TIP
If the presentation is not product focused, simply delete this. Do NOT insert the Red Hat logo here. If this is a product-focused deck, right click on the logo and using “Replace Image” insert the product logo of your choice. After replacing the image, right click and select “Reset Image”. Adjust spacing as needed.

Presentation title should not exceed two lines

Optional subheading

Presenter’s Name
Title

Presenter’s Name
Title

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat

Presentation title should not exceed three lines

Optional subheading Lorem ipsum dolor sit amet consectetuer adipiscing elit sed diam

Presenter's Name
Title

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/in/karelhala

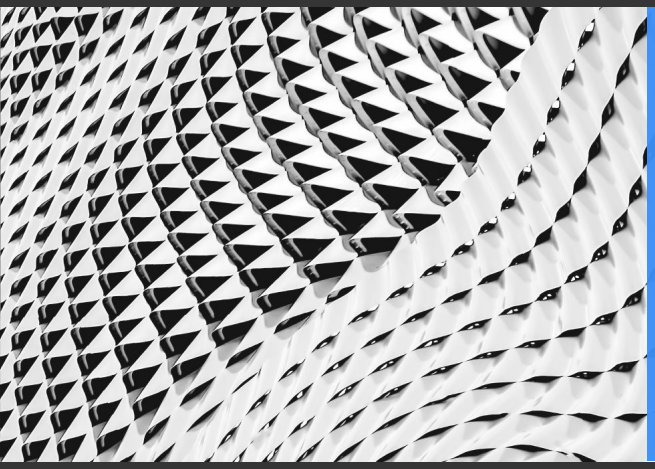
linkedin.com/in/coufaltom

github.com/karelhala

github.com/tumido

Red Hat

Divider title limit to two lines

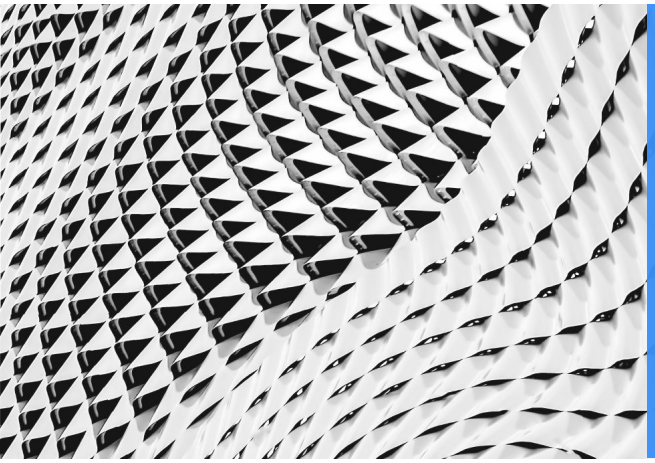


QUICK TIP
Try right clicking on the photo and using “Replace Image” to insert your own photo. You are also welcome to use this photo.

Optional supporting copy. Lorem ipsum dolor sit amet, consectetuer adipis elit, sed diam nonummy nibh euismod tincidunt ut laoreet. magna aliquam.

Red Hat

Divider title limit to two lines

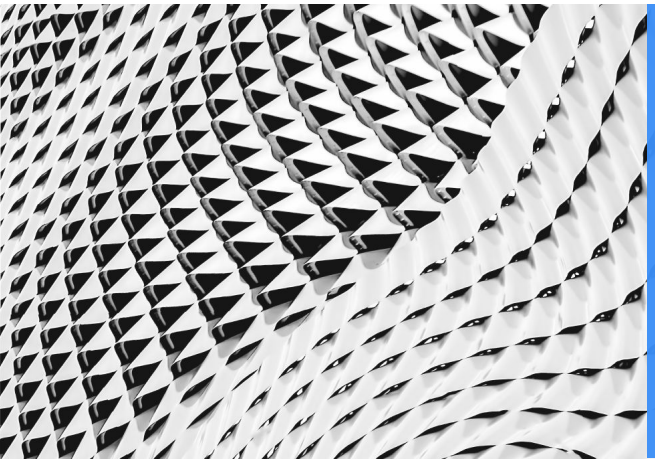


QUICK TIP
Try right clicking on the photo and using "Replace Image" to insert your own photo. You are also welcome to use this photo.

Optional supporting copy. Lorem ipsum dolor sit amet, consectetuer adipis elit, sed diam nonummy nibh euismod tincidunt ut laoreet. magna aliquam.

Red Hat

Divider title limit to two lines



QUICK TIP
Try right clicking on the photo and using “Replace Image” to insert your own photo. You are also welcome to use this photo.

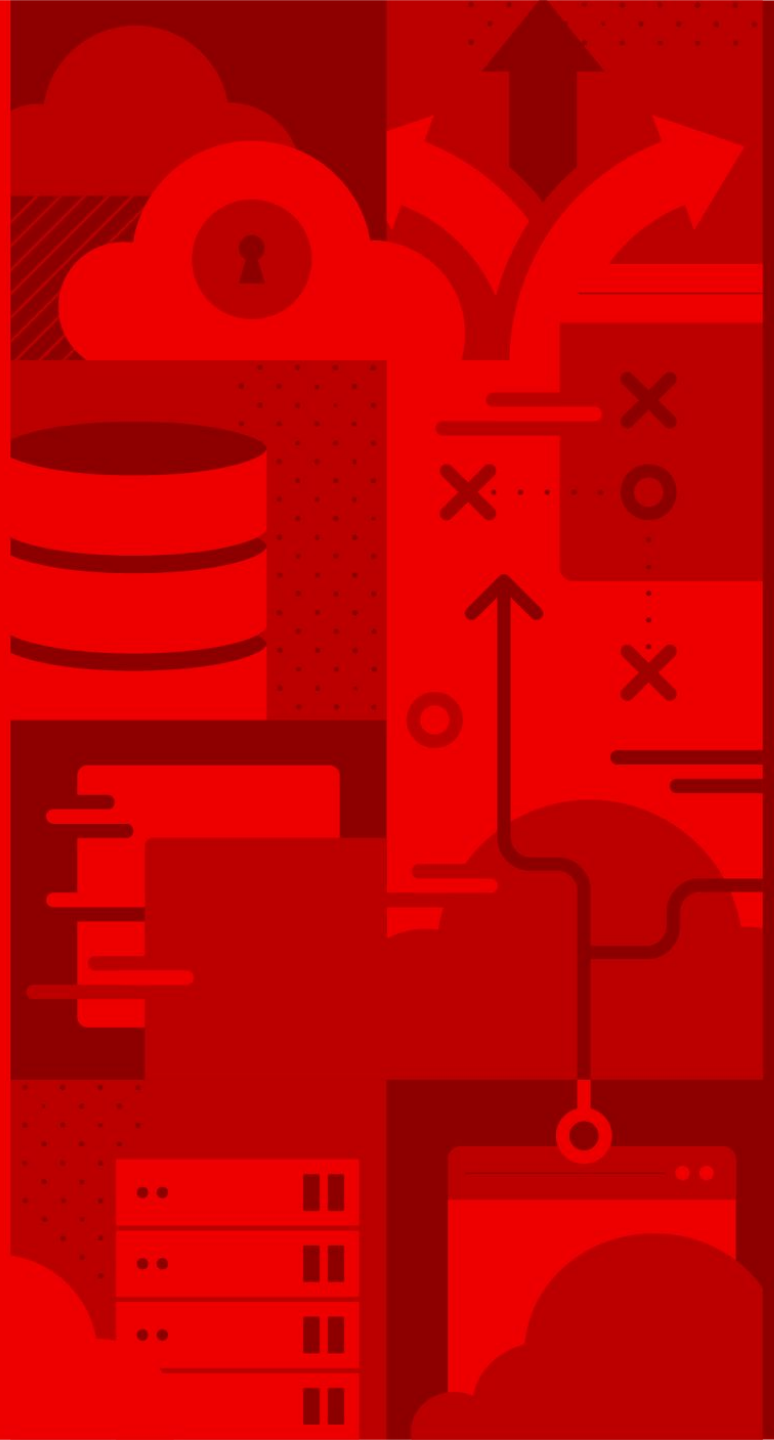
Optional supporting copy. Lorem ipsum dolor sit amet, consectetuer adipis elit, sed diam nonummy nibh euismod tincidunt ut laoreet. magna aliquam.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipisc elit sed dia nibh?

Lorem ipsum dolor sit amet, consectetuer adipisc elit sed dia nibh?

Red Hat



Lorem ipsum dolor sit amet, consectetuer adipisc elit?

Red Hat

This section includes:

Agenda slide templates

Content slide templates

Quote slide templates

What we'll discuss today

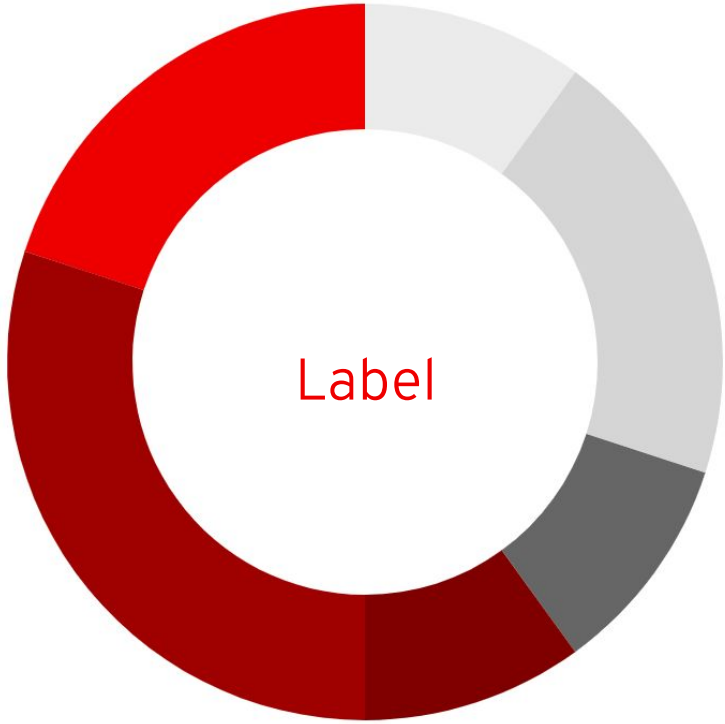
Topic	Topic	Topic
Topic	Topic	Topic
Topic	Topic	Topic
Topic	Topic	Topic
Topic	Topic	Topic

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullam.



00% Lorem ipsum dolor sit amet

00% Lorem ipsum dolor sit amet

00% Lorem ipsum dolor sit amet

00% Lorem ipsum dolor sit amet

00% Lorem ipsum dolor sit amet

00% Lorem ipsum dolor sit amet

Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
To edit a chart, select it and click the dropdown arrow in the top right. Select "Open source." The legend will need to be updated manually.

Red Hat

What we'll discuss today

Topic	Topic	Topic
Topic	Topic	Topic
Topic	Topic	Topic
Topic	Topic	Topic
Topic	Topic	Topic

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

QUICK TIP

Insert image in this designated area, deleting the shaded background. Keep the left and right margins clear to maintain the open feel in accordance with the brand.

If no subheading is needed, delete the subheading text, and the content can extend into the lighter shaded area.

Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP

Insert image in this designated area, deleting the shaded background. Keep the left, right, top, and bottom margins clear to maintain the open feel in accordance with the brand.

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus. Etiam interdum nunc non venenatis rutrum. Phasellus venenatis, sem ac vulputate facilisis, lacus augue vehicula quam. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus.

- Etiam interdum nunc non venenatis rutrum
- Phasellus venenatis sem ac vulputate facilisis
- Aenean eleifend laoreet eros eu molestie arcu tempus
- Quisque vitae nisl accumsan aliquet

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus. Etiam interdum nunc non venenatis rutrum. Phasellus venenatis, sem ac vulputate facilisis, lacus augue vehicula quam. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus.

- Etiam interdum nunc non venenatis rutrum
- Phasellus venenatis sem ac vulputate facilisis
- Aenean eleifend laoreet eros eu molestie arcu tempus
- Quisque vitae nisl accumsan aliquet

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus. Etiam interdum nunc non venenatis rutrum. Phasellus venenatis, sem ac vulputate facilisis, lacus augue vehicula quam. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus.

- Etiam interdum nunc non venenatis rutrum
- Phasellus venenatis sem ac vulputate facilisis
- Aenean eleifend laoreet eros eu molestie arcu tempus
- Quisque vitae nisl accumsan aliquet

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

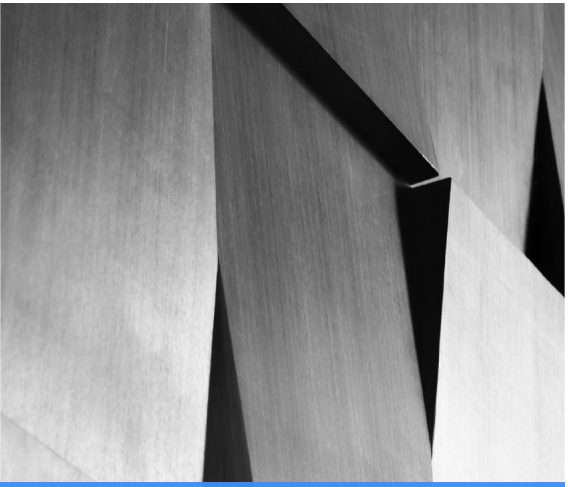
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus. Etiam interdum nunc non venenatis.

- Etiam interdum nunc non venenatis rutrum
- Phasellus venenatis sem ac vulputate facilisis
- Aenean eleifend laoreet eros eu molestie arcu tempus
- Quisque vitae nisl accumsan aliquet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus. Etiam interdum nunc non venenatis.

- Etiam interdum nunc non venenatis rutrum
- Phasellus venenatis sem ac vulputate facilisis
- Aenean eleifend laoreet eros eu molestie arcu tempus
- Quisque vitae nisl accumsan aliquet

Source:
Insert source data here
Insert source data here
Insert source data here



QUICK TIP
Try right clicking on the photo and using “Replace Image” to insert your own photo. You are also welcome to use this photo.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus. Etiam interdum nunc non venenatis rutrum. Phasellus venenatis, sem ac vulputate facilisis, lacus augue vehicula quam.

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eleifend laoreet eros, eu molestie arcu tempus ac. Quisque vitae nisl accumsan, aliquet est et, varius purus. Etiam interdum nunc non venenatis rutrum. Phasellus venenatis, sem ac vulputate facilisis, lacus augue vehicula quam, pellentesque pulvinar elit magna posuere magna.

QUICK TIP
Try right clicking on the photo and using “Replace Image” to insert your own photo. You are also welcome to use this photo.

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

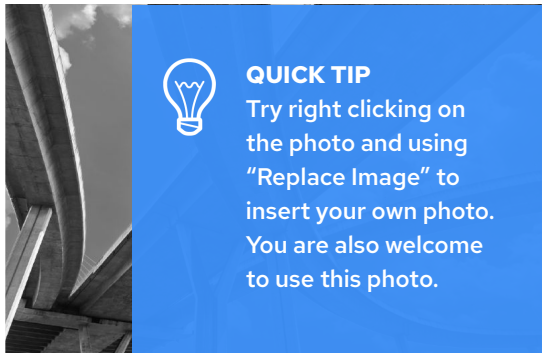
Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh mod tincidunt.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh mod tincidunt.



Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh mod tincidunt.

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh mod tincidunt.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh mod tincidunt.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh mod tincidunt.

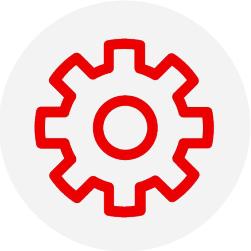
Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Try right clicking on the icon and using "Replace Image" to insert your own icons.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna



Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

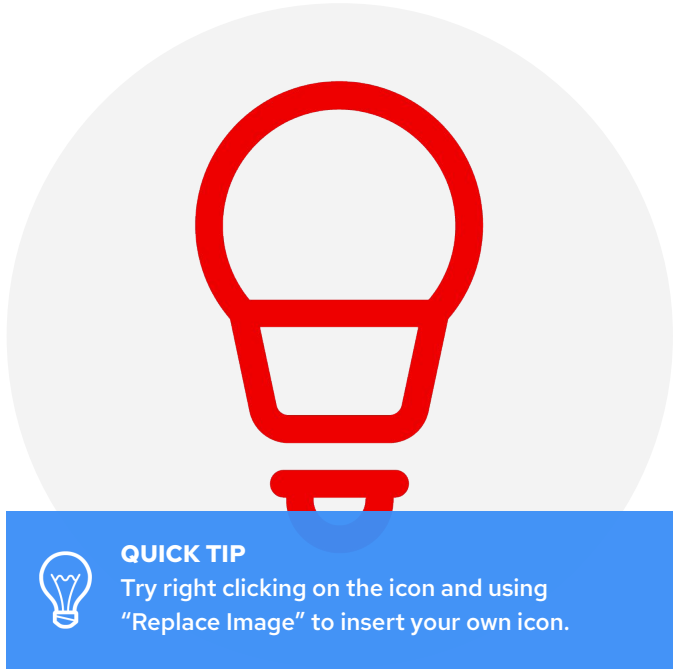
Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Try right clicking on the icon and using "Replace Image" to insert your own icons.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.



Body headline

Lorem ipsum dolor sit amet, consectet adipiscing elit sed dia.

Body headline

Lorem ipsum dolor sit amet, consectet adipiscing elit sed dia.

Body headline

Lorem ipsum dolor sit amet, consectet adipiscing elit sed dia.

Body headline

Lorem ipsum dolor sit amet, consectet adipiscing elit sed dia.

Source:
Insert source data here
Insert source data here
Insert source data here

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt.

Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Try right clicking on the icon and using "Replace Image" to insert your own icons.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

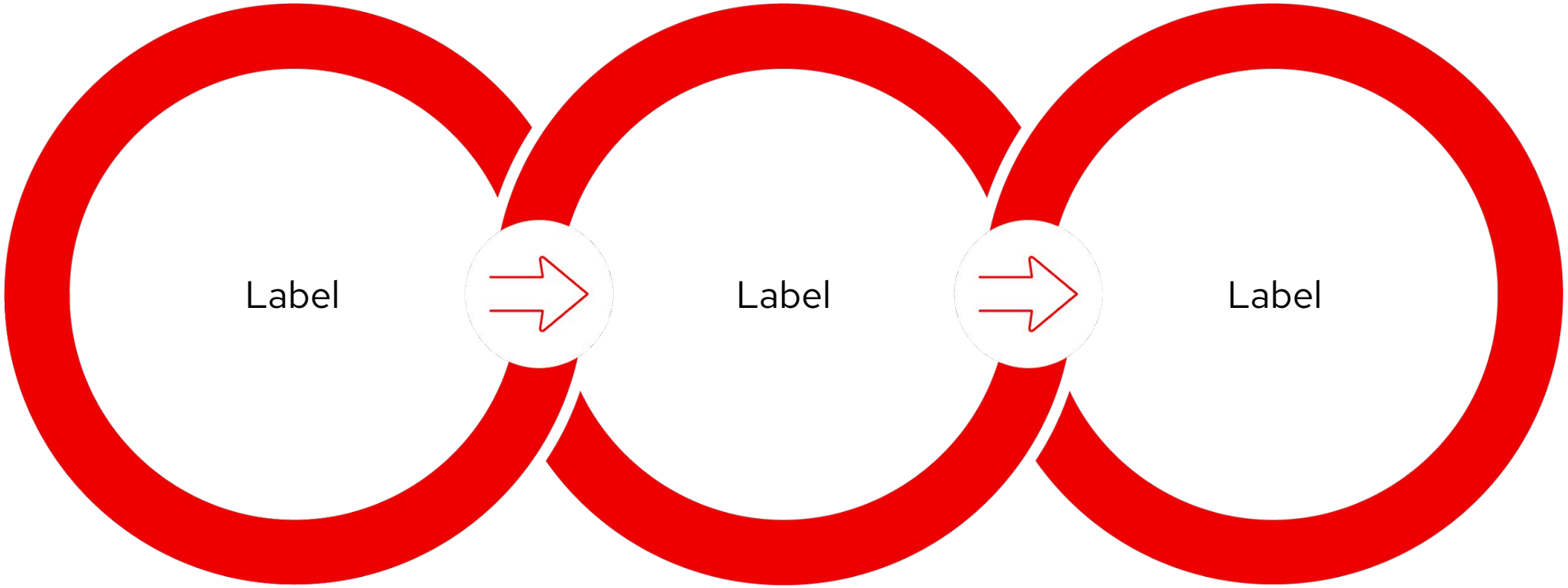
Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Try right clicking on the icon and using “Replace Image” to insert your own icons.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.



Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

Lorem ipsum dolor sit amet, consectetuer adipiscin elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore

Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Using a photo with the large quote is optional. Try right clicking on the photo and using “Replace Image” to insert your own photo.

“Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis id auctor dui. Ut neque sem, convallis sit amet ultrices et, facilisis vestibulum ligula. Donec euismod elementum erat vitae fermentum. Mauris hendrerit maximus bibendum.”

John Doe
CTO, Acme Unlimited

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

"Lorem ipsum dolor sit amet, consectetuer adipis elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullam."

John Doe
CTO, Acme Unlimited

"Lorem ipsum dolor sit amet, consectetuer adipis elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullam."

John Doe
CTO, Acme Unlimited

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

John Doe
CTO, Acme Unlimited

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

John Doe
CTO, Acme Unlimited

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

John Doe
CTO, Acme Unlimited

Source:
Insert source data here
Insert source data here
Insert source data here

This section includes:

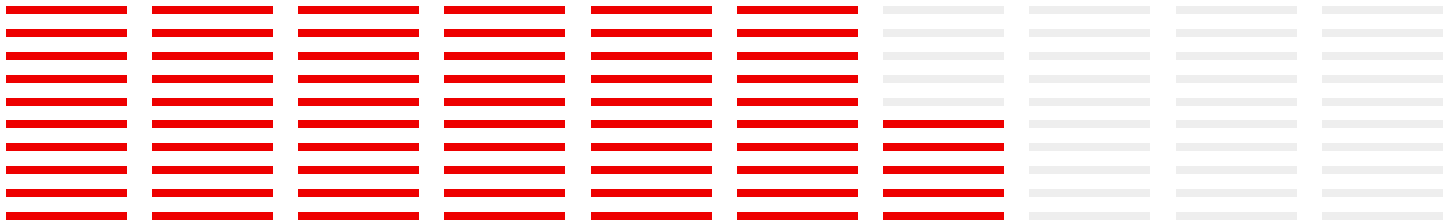
Data slide templates

Table slide templates

Timeline slide templates

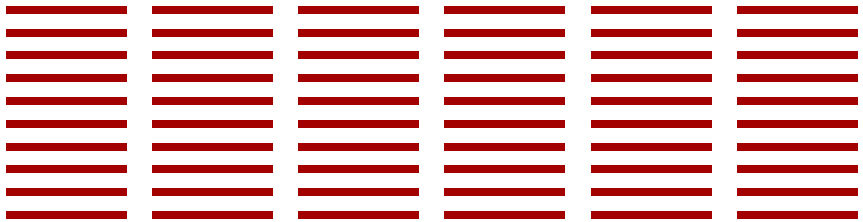
Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.



65%

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.



QUICK TIP
Use darker shades of Red Hat red to differentiate between data points.

82%

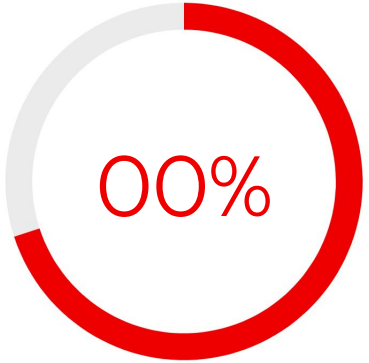
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

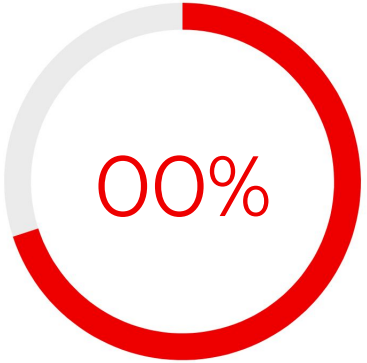
Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullam.



Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.



Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

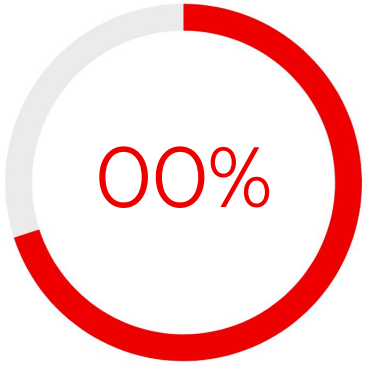
Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
To edit a chart, select it and click the dropdown arrow in the top right. Select "Open source." The percentage will need to be updated manually.

Red Hat

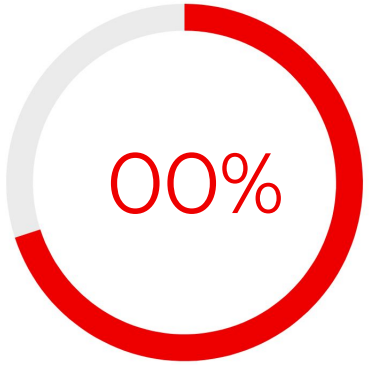
Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.



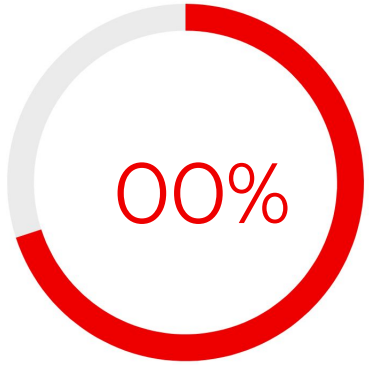
Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.



Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.



Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
To edit a chart, select it and click the dropdown arrow in the top right. Select "Open source." The percentage will need to be updated manually.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullam.

000

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

000

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

Source:
Insert source data here
Insert source data here
Insert source data here

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

000

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

000

Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

000

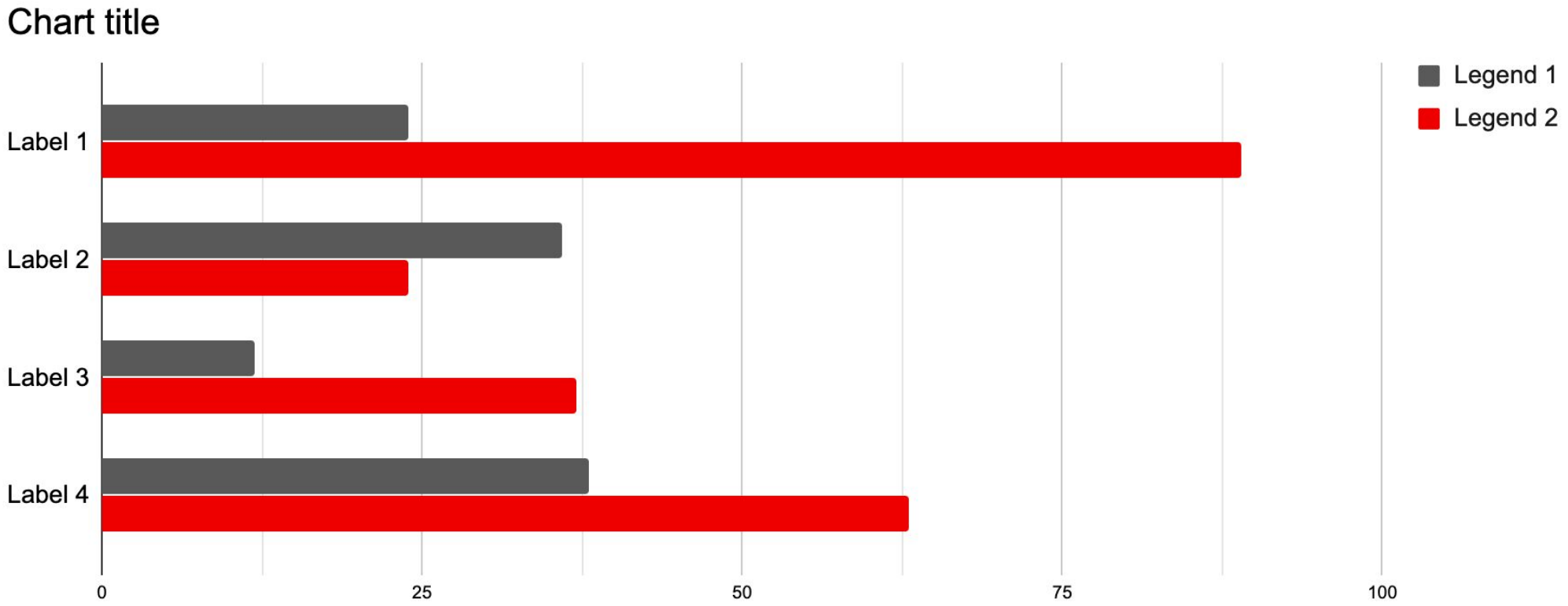
Body headline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.



Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
To edit a chart, select it and click the dropdown arrow in the top right. Select "Open source."

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

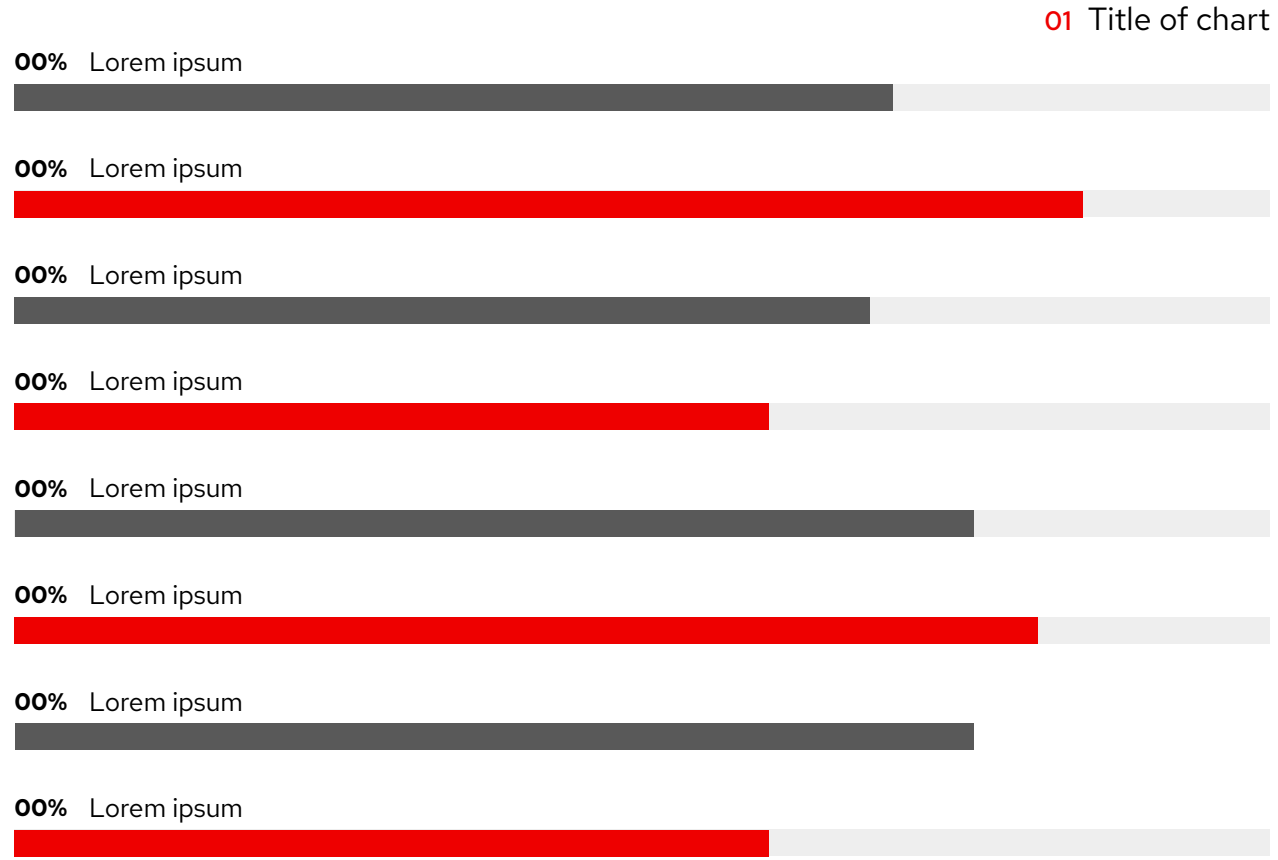
Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Use the "Width Scale" in the "Size & Position" pane of the format options panel to adjust the percentage of the bar.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna



Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet

Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Use the "Width Scale" in the "Size & Position" pane of the format options panel to adjust the percentage of the bar.

Red Hat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna

01 Title of table

	Column header two lines maximum	Column header two lines maximum	Column header two lines maximum
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.

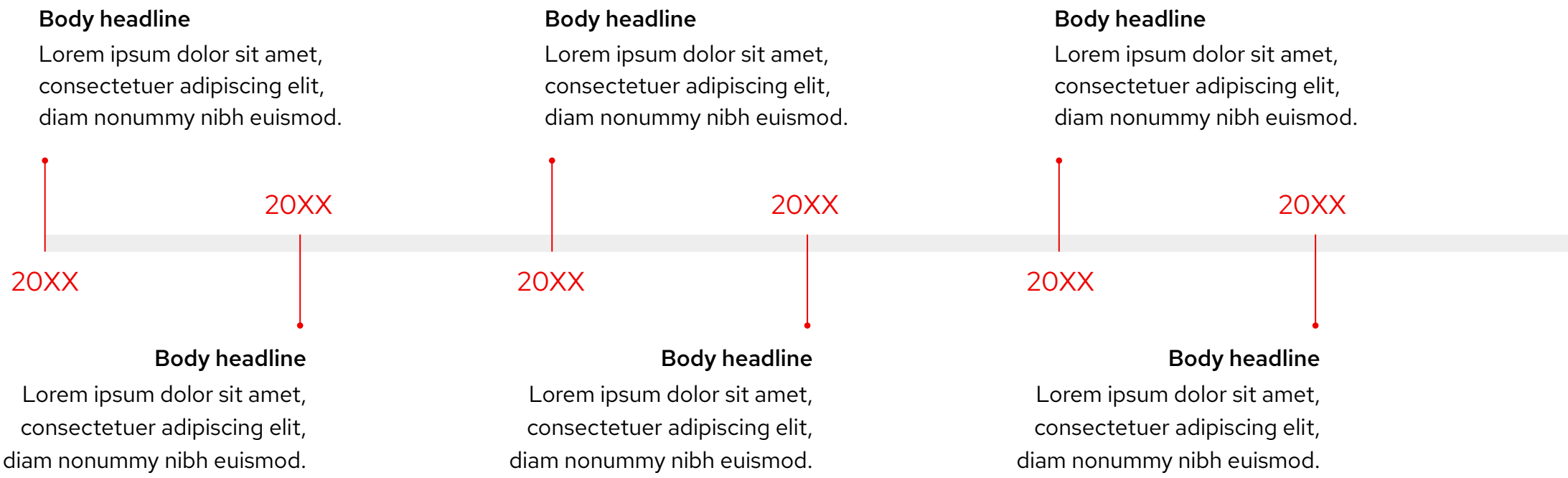
01 Title of table

	Column header two lines maximum	Column header two lines maximum	Column header two lines maximum	Column header two lines maximum	Column header two lines maximum
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines
Row header with two lines maximum	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines	Body cell should be limited to two lines

Source:
Insert source data here
Insert source data here
Insert source data here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Duis vel mauris aliquet, aliquam velit eu, euismod lorem.



Source:
Insert source data here
Insert source data here
Insert source data here

QUICK TIP
Try right clicking on the photo and using “Replace Image” to insert your own photo. You are also welcome to use this photo.

Heading

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sedie diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip.

Source:
Insert source data here
Insert source data here
Insert source data here

Red Hat

20XX

Heading

Lorem ipsum dolor sit amet, consectet adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolor aliquam.

QUICK TIP
Try right clicking on the photo and using “Replace Image” to insert your own photo. You are also welcome to use this photo.

Heading

Lorem ipsum dolor sit amet, consectet adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolor aliquam.

20XX

Heading

Lorem ipsum dolor sit amet, consectet adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolor aliquam.

Source:
Insert source data here
Insert source data here
Insert source data here

Red Hat

Heading

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sedie diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip.

QUICK TIP
Try right clicking on the photo and using “Replace Image” to insert your own photo. You are also welcome to use this photo.

Source:
Insert source data here
Insert source data here
Insert source data here